Cognitive Mapping and Planning for Visual Navigation Supplementary Material

Saurabh Gupta[1,2] James Davidson[2] Sergey Levine[1,2] Rahul Sukthankar[2] Jitendra Malik[1,2]
[1]UC Berkeley [2]Google
[1]{sgupta, svlevine, malik}@eecs.berkeley.edu, [2]{jcdavidson, sukthankar}@google.com

A1. Backward Flow Field ρ from Egomotion

Consider a robot that rotates about its position by an angle θ and then moves t units forward. Corresponding points p in the original top-view and p' in the new top-view are related to each other as follows (R_θ is a rotation matrix that rotates a point by an angle θ):

$$p' = R_\theta^t p - t \text{ or } p = R_\theta(p' + t) \quad (1)$$

Thus given the egomotion θ and t, for each point in the new top-view we can compute the location in the original top-view from which it came from.

A2. Mapper Performance in Isolation

To demonstrate that our proposed mapper architecture works we test it in isolation on the task of free space prediction. We consider the scenario of an agent rotating about its current position, and the task is to predict free space in a 3.20 meter neighborhood of the agent. We only provide supervision for this experiment at end of the agents rotation. Figure A1 illustrates what the mapper learns. Observe that our mapper is able to make predictions where no observations are made. We also report the mean average precision for various versions of the mapper Table A1 on the test set (consisting of 2000 locations from the testing environment). We compare against an analytic mapping baseline which projects points observed in the depth image into the top view (by back projecting them into space and rotating them into the top-down view).

A3. Additional Experiments

Additional experiment on an internal Matterport dataset. We also conduct experiments on an internal Matterport dataset consisting of 41 scanned environments. We train on 27 of these environments, use 4 for validation and test on the remaining 10. We show results for the 10 test environments in Figure A2. We again observe that CMP consistently outperforms the 4 frame reactive baseline and LSTM.

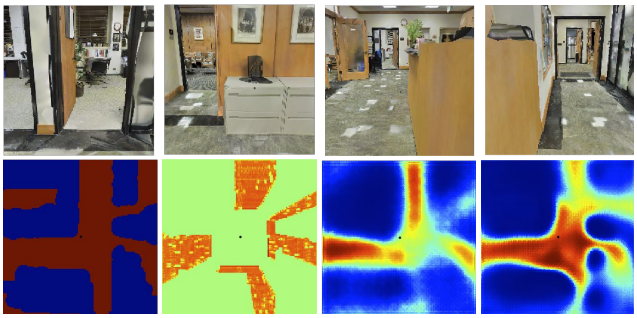

Figure A1: Output Visualization for Mapper trained for Free Space Prediction: We visualize the output of the mapper when directly trained for task of predicting free space. We consider the scenario of an agent rotating about its current position, the task is to predict free space in a 3.20 meter neighborhood of the agent, supervision for this experiment at end of the agents rotation. The top row shows the 4 input views. The bottom row shows the ground truth free space, predicted free space by analytically projecting the depth images, learned predictor using RGB images and learned predictor using depth images. Note that the learned approaches produce more complete output and are able to make predictions where no observations were made.

Method	Modality	CNN Architecture	Free Space Prediction AP
Analytic Projection	depth	-	56.1
Learned Mapper	RGB	ResNet-50	74.9
Learned Mapper	depth	ResNet-50 Random Init.	63.4
Learned Mapper	depth	ResNet-50 Init. using [1]	78.4

Table A1: Mapper Unit Test: We report average precision for free space prediction when our proposed mapper architecture is trained directly for the task of free space prediction on a test set (consisting of 2000 locations from the testing environment). We compare against an analytic mapping baseline which projects points observed in the depth image into the top view (by back projecting them into space and rotating them into the top-down view).

Work done when S. Gupta was an intern at Google.

Project website with videos: https://sites.google.com/view/cognitive-mapping-and-planning/.

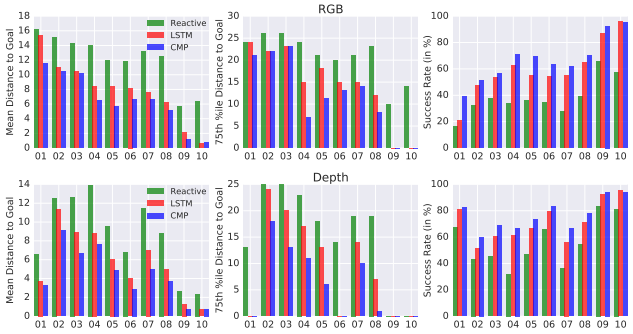

Figure A2: We report the mean distance to goal, 75th percentile distance to goal (lower is better) and success rate (higher is better) for Reactive, LSTM and CMP based agents on different test environments from an internal dataset of Matterport scans. We show performance when using RGB images (top row) and depth images (bottom row) as input. We note that CMP consistently outperforms Reactive and LSTM based agents.

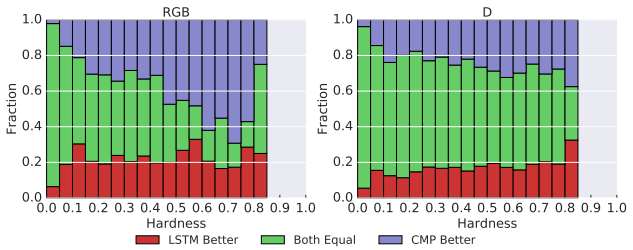

Figure A3: We show how performance of LSTM and CMP compare across geometric navigation tasks of different hardness. We define hardness as the gap between the ground truth and heuristic (Manhattan) distance between the start and goal, normalized by the ground truth distance. For each range of hardness we show the fraction of cases where LSTM gets closer to the goal (LSTM Better), both LSTM and CMP are equally far from the goal (Both Equal) and CMP gets closer to goal than LSTM (CMP Better). We show results when using RGB images as input (left plot) and when using Depth images as input (right plot). We observe that CMP is generally better across all values of hardness, but for RGB images it is particularly better for cases with high hardness.

Method	Mean		75th %ile		Success %age	
	RGB	Depth	RGB	Depth	RGB	Depth
Geometric Task						
Initial	25.3	25.3	30	30	0.7	0.7
No Image LSTM	20.8	20.8	28	28	6.2	6.2
CMP						
Full model	7.7	4.8	14	1	62.5	78.3
Single-scale planner	7.9	4.9	12	1	63.0	79.5
Shallow planner	8.5	4.8	16	1	58.6	79.0
Analytic map	-	8.0	-	14	-	62.9

Table A2: Ablative Analysis for CMP: We follow the same experimental setup as used for table in the main text. See text for details.

Ablations. We also present performance of ablated versions of our proposed method in Table A2.

Single Scale Planning. We replace the multi-scale planner with a single-scale planner. This results in slightly better performance but comes at the cost of increased planning cost.

	Mean			75th %ile			Success Rate (in %)		
	Init.	LSTM	CMP	Init.	LSTM	CMP	Init.	LSTM	CMP
Far away goal (maximum 64 steps away)									
Run for 79 steps	47.2	15.2	11.9	58	29	19.2	0.0	58.4	66.3
Run for 159 steps	47.2	12.5	9.3	58	19	0	0.0	69.0	78.5
Generalization									
Train on 1 floor	25.3	8.9	7.0	30	18	10	0.7	58.9	67.9
Transfer from IMD	25.3	11.0	8.5	30	21	15	0.7	48.6	61.1

Table A3: We report additional comparison between best performing models. See text for details.

No Planning. We swap out the planner CNN with a shallower CNN. This also results in drop in performance specially for the RGB case as compared to the full system which uses the full planner.

Analytic Mapper. We also train a model where we replace our learned mapper for an analytic mapper that projects points from the depth image into the overhead view and use it with a single scale version of the planner. We observe that this analytic mapper actually works worse than the learned one thereby validating our architectural choice of learning to map.

Additional comparisons between LSTM and CMP. We also report additional experiments on the Stanford S3DIS dataset to further compare the performance of the LSTM baseline with our model in the most competitive scenario where both methods use depth images. These are reported in Table A3. We first evaluate how well do these models perform in the setting when the target is much further away (instead of sampling problems where the goal is within 32 time steps we sample problems where the goal is 64 times steps away). We present evaluations for two cases, when this agent is run for 79 steps or 159 steps (see 'Far away goal' rows in Table A3). We find that both methods suffer when running for 79 steps only, because of limited time available for back-tracking, and performance improves when running these agents for longer. We also see a larger gap in performance between LSTM and CMP for both these test scenarios, thereby highlighting the benefit of our mapping and planning architecture.

We also evaluate how well these models generalize when trained on a single scene ('Train on 1 scene'). We find that there is a smaller drop in performance for CMP as compared to LSTM. We also found CMP to transfer from internal Matterport dataset to the Stanford S3DIS Dataset slightly better ('Transfer from internal dataset').

We also study how performance of LSTM and CMP compares across geometric navigation tasks of different hardness in Figure A3. We define hardness as the gap between the ground truth and heuristic (Manhattan) distance between the start and goal, normalized by the ground truth distance. For each range of hardness we show the fraction of

Method	Mean		50th %ile		75th %ile		Success %age	
	RGB	Depth	RGB	Depth	RGB	Depth	RGB	Depth
ST: Aggregate								
Initial	16.2	16.2	17	17	25	25	11.3	11.3
Reactive	14.2	14.2	14	13	22	23	23.4	22.3
LSTM	13.5	13.4	13	14	20	23	23.5	27.2
CMP	11.3	11.0	11	9	18	19	34.2	40.0
ST: Chair								
Initial	16.5	16.5	17	17	24	24	9.9	9.9
Reactive	13.6	13.6	13	12	21	22	22.0	16.9
LSTM	13.7	14.5	13	15	20	23	17.9	23.1
CMP	11.0	10.3	8	6	18	18	32.8	40.6
ST: Door								
Initial	16.0	16.0	16	16	24	24	11.9	11.9
Reactive	14.5	14.4	15	13	24	23	24.8	26.2
LSTM	13.7	13.4	14	14	21	23	26.9	28.9
CMP	10.6	11.8	13	10	17	20	38.3	40.3
ST: Table								
Initial	16.3	16.3	17	17	25	25	11.7	11.7
Reactive	14.3	14.4	15	16	22	22	21.9	20.7
LSTM	12.9	11.8	13	11	20	19	23.6	28.9
CMP	13.0	10.0	12	9	22	17	26.4	38.2
ST + Markers: Aggregate								
Initial	16.2	16.2	17	17	25	25	11.3	11.3
Reactive	13.6	12.6	14	10	23	23	29.1	36.7
LSTM	12.4	10.4	12	10	22	17	35.0	40.8
CMP	11.1	9.9	9	1	20	18	44.0	55.2

Table A4: Navigation Results for Semantic Task: We report the mean distance to goal location, 50th percentile, 75th percentile distance to goal and success rate after executing the policy for 39 time steps for the semantic task. We report the aggregate performance (across all different categories) and performance for each category independently. The bottom part (ST + Markers) presents results for the case where a distinctive marker is placed at the location of the object to factor out bottlenecks in performance due to difficulty in recognizing target objects.

cases where LSTM gets closer to the goal (LSTM Better), both LSTM and CMP are equally far from the goal (Both Equal) and CMP gets closer to goal than LSTM (CMP Better). We observe that CMP is generally better across all values of hardness, but for RGB images it is particularly better for cases with high hardness.

Semantic Task. Table A4 reports per category performance for the semantic task. We also report an experiment where objects locations are explicitly marked by an easily identifiable 'marker' (a floating cube at the location of the chair, reported as 'ST + Markers' in Table A4). We found this to boost performance suggesting incorporating appearance models in the form of object detectors from large-scale external datasets will improve performance for semantic tasks.

Visualizations. To better understand the representation learned by the mapper, we train readout functions on the learned mapper representation to predict free space. Figure A4 visualizes these readout functions at two time steps from an episode as the agent moves. As expected, the representation output by the mapper carries information about free space in the environment. Readouts are generally better at finer scales. Finally, Figure A5 visualizes a 1 channel projection of the value map for the single scale version of our model at five time steps from an episode.

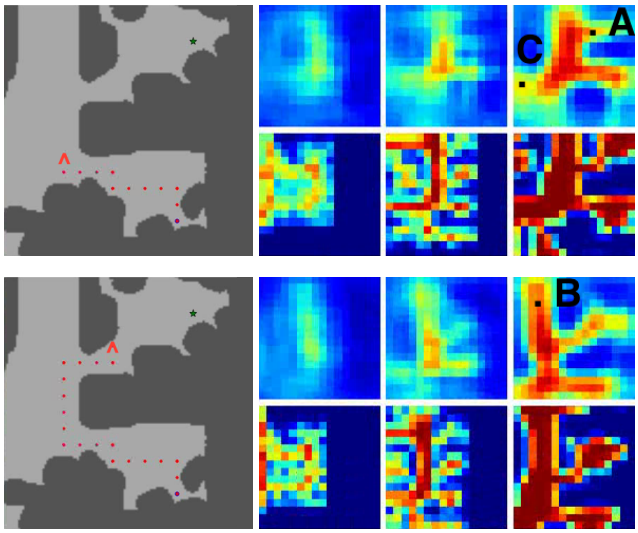

Figure A4: We visualize the output of the map readout function trained on the representation learned by the mapper (see text for details) as the agent moves around. The two rows show two different time steps from an episode. For each row, the gray map shows the current position and orientation of the agent (red ∧), and the locations that the agent has already visited during this episode (red dots). The top three heatmaps show the output of the map readout function and the bottom three heatmaps show the ground truth free space at the three scales used by CMP (going from coarse to fine from left to right). We observe that the readout maps capture the free space in the regions visited by the agent (room entrance at point A, corridors at points B and C).

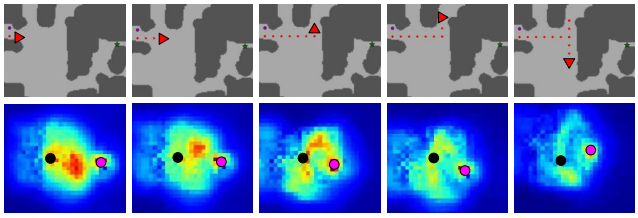

Figure A5: We visualize the value function for five snapshots for an episode for the single scale version of our model. The top row shows the agent's location and orientation with a red triangle, nodes that the agent has visited with red dots and the goal location with the green star. Bottom row shows a 1 channel projection of the value maps (obtained by taking the channel wise max) and visualizes the agent location by the black dot and the goal location by the pink dot. Initially the agent plans to go straight ahead, as it sees the wall it develops an inclination to turn left. It then turns into the room (center figure), planning to go up and around to the goal but as it turns again it realizes that that path is blocked (center right figure). At this point the value function changes (the connection to the goal through the top room becomes weaker) and the agent approaches the goal via the downward path.

A4. Experimental Testbed Details

We pre-processed the meshes to compute space traversable by the robot. Top views of the obtained traversable space are shown in Figure A6 (training and validation) and Figure A7 (testing) and indicate the complexity of the environments we are working with and the differences in layouts between the training and testing environments. Recall that robot's action space $\mathcal{A}_{x,\theta}$ consists of macro-actions. We pick θ to be $\pi/2$ which allows us to pre-compute the set of locations (spatial location and orientation) that the robot can visit in this traversable space. We also precompute a directed graph $\mathcal{G}_{x,\theta}$ consisting of this set of locations as nodes and a connectivity structure based on the actions available to the robot.

Our setup allows us to study navigation but also enables us to independently develop and design our mapper and planner architectures. We developed our mapper by studying the problem of free space prediction from sequence of first person view as available while walking through these environments. We developed our planner by using the ground truth top view free space as 2D mazes to plan paths through. Note that this division was merely done to better understand each component, the final mapper and planner are trained jointly and there is no restriction on what information gets passed between the mapper and the planner.

A5. Discussion

In this paper, we introduced a novel end-to-end neural architecture for navigation in novel environments. Our architecture learns to map from first-person viewpoints and uses a planner with the learned map to plan actions for navigating to different goals in the environment. Our experiments demonstrate that such an approach outperforms other direct methods which do not use explicit mapping and planning modules. While our work represents exciting progress towards problems which have not been looked at from a learning perspective, a lot more needs to be done for solving the problem of goal oriented visual navigation in novel environments.

A central limitations in our work is the assumption of perfect odometry. Robots operating in the real world do not have perfect odometry and a model that factors in uncertainty in movement is essential before such a model can be deployed in the real world.

A related limitation is that of building and maintaining metric representations of space. This does not scale well for large environments. We overcome this by using a multi-scale representation for space. Though this allows us to study larger environments, in general it makes planning more approximate given lower resolution in the coarser scales which could lead to loss in connectivity information. Investigating representations for spaces which do not suffer from such limitations is important future work.

In this work we have exclusively used DAGGER for training our agents. Though this resulted in good results, it suffers from the issue that the optimal policy under an expert may be unfeasible under the information that the agent currently has. Incorporating this in learning through guided

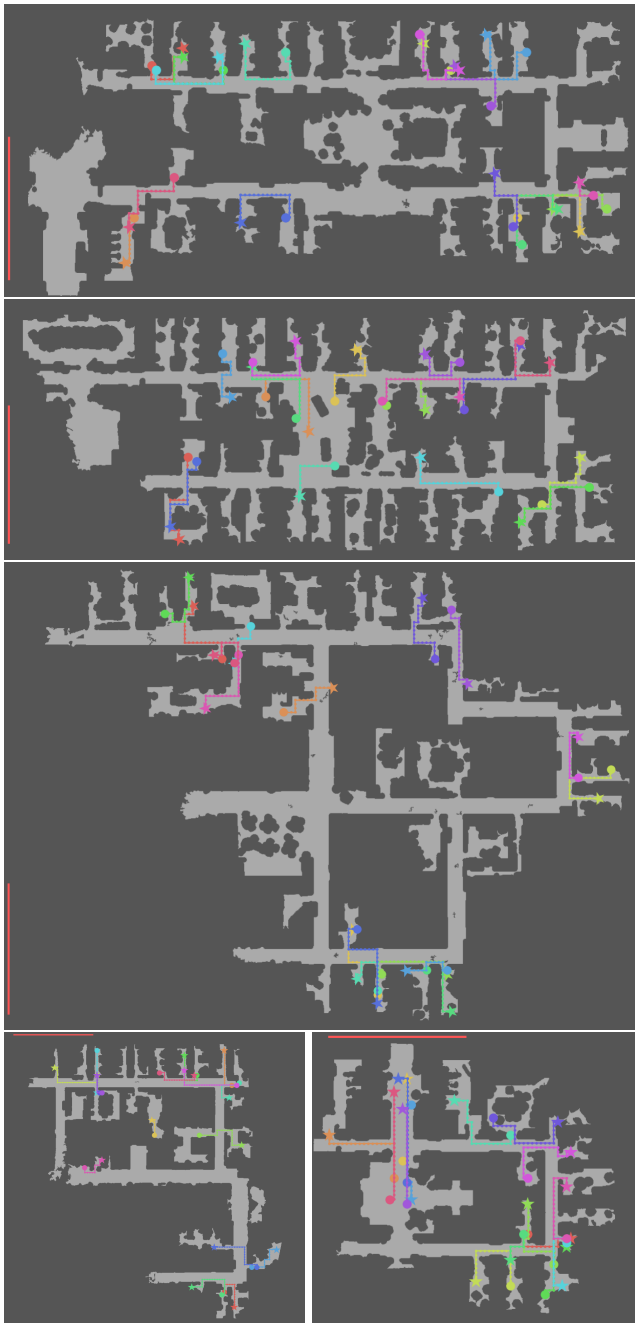

Figure A6: Maps for *area1*, *area6*, *area51*, *area52* and *area3*. Light area shows traversable space. Red bar in the corner denotes a length of 32 units (12.80 metres). We also show some example geometric navigation problems in these environments, the task is to go from the circle node to the star node.

Figure A7: Map for *area4*. This floor was used for testing all the models. Light area shows traversable space. Red bar in the corner denotes a length of 32 units (12.80 metres). We also show some example geometric navigation problems in these environments, the task is to go from the circle node to the star node.

policy search or reinforcement learning may lead to better performance specially for the case when the goal is not specified geometrically.

References

[1] S. Gupta, J. Hoffman, and J. Malik. Cross modal distillation for supervision transfer. In *CVPR*, 2016. 1